Intrusion Detection using Honeypot and Support Vector Machine Classifier

Kanchan Shendre

Roll. 213CS2166

under the supervision of

Prof. Ratnakar Dash

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769008, India

Intrusion Detection using Honeypot and Support Vector Machine Classifier

Dissertation submitted in

MAY 2015

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

by

Kanchan Shendre

(Roll. 213CS2166)

under the supervision of

Prof. Ratnakar Dash

Department of Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela – 769 008, India

Computer Science and Engineering

National Institute of Technology Rourkela

Rourkela-769 008, India. www.nitrkl.ac.in

May 30, 2015

Certificate

This is to certify that the work in the project entitled *Intrusion Detection System with Honeypot and Multiclass Support Vector Machine Classifier* by *Kanchan Shendre* is a record of their work carried out under my supervision and guidance in partial fulfillment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering*.

Dr. Ratnakar Dash
Associate Professor
Department of CSE
NIT, Rourkela

Acknowledgment

First of all, I would like to express my deep sense of respect and gratitude towards my supervisor Prof. Ratnakar Dash, who has been the guiding force behind this work. I want to thank him for introducing me to the field of Service-Oriented Architecture and giving me the opportunity to work with him. His undivided faith in this topic and ability to bring out the best of analytical and practical skills in people has been invaluable in tough periods. Without his valuable advice and assistance, it would not have been possible for me to complete this thesis. I am greatly indebted to him for his constant encouragement and valuable advice in every aspect of my academic life. I consider it my good fortune to have got an opportunity to work with such a wonderful person.

I would also like to thank all faculty members, Ph.D. scholars, my seniors and juniors and all colleagues to provide me their regular suggestions and encouragements during the whole work. I am very grateful to Santosh Sir who helped me in every possible way during the entire academic year.

At last but not the least I am in debt to my family to support me regularly during my hard times.

I wish to thank all faculty members and secretarial staff of the CSE Department for their sympathetic cooperation.

Kanchan Shendre

Abstract

The rapid growth of internet and web based applications has given rise to the number of attacks on the network. The way the attacker attacks the system differs from one attacker to the other. The sequence of attack or the signature of an attacker should be stored, analyzed and used to generate rules for mitigating future attack attempts. We have deployed honeypot to record the activities of the attacker. While the attacker prepares for an attack, the IDS redirects him to the honeypot. We make the attacker believe that he is working with the actual system. The activities related to the attack are recorded by the honeypot by interacting with the intruder. The recorded activities are analyzed by the network administrator, and the rule database is updated. As a result, we improve the detection accuracy and security of the system using honeypot without any loss or damage to the original system. As the number of threats to the information is increasing, there is a need for a powerful intrusion detection system that can actually fulfil the requirement of security against the threat. This type of security can be achieved by identifying the particular type of attack. The classification of attack activities ensures the efficient countermeasure for the attack. The work focuses on the classification of attack using multiclass support vector machine approach. The support vector machine is used for binary classification. This approach is extended to the multiclass classification of attack with improved accuracy of classification. We have used three benchmark datasets for training and testing purpose: KDD corrected dataset, NSLKDD dataset, Gure KDD dataset. We have also compared the results with existing work. The evaluation gives better accuracy for detection of attack than the existing approach. The evaluation provides better accuracy for detection of attack than the existing approach.

***Keywords*:** Honeypot, IDS, Threat, KDD corrected dataset, NSLKDD dataset, Gure KDD dataset

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The highly integrated electronic world is an effect of technological development over decades. The number of malicious activities and attacks are also growing beside the advances in security against threats. To cope with this situation, various attempts are made to control the attack activities. There is a need to improve and innovate different techniques for the detection of intrusion against the crucial as well as an enormous amount of information [1].

Intrusion Detection

Intrusion is an unauthorised access to the system with the intent of doing theft of information or harms the system. The act of detecting intrusions, monitoring the incidents occurring in the computer system, the suspicious or unusual activities, taking place in the system, which can be the possible attack, is known as intrusion detection [1].

Intrusion Detection System

The software or hardware system that monitors the information for attack and identifies the possible events carried out by the intruder, attempts to eliminate them

and prepares a record to send it to the administrators in a real-time situation. The Intrusion detection system is different from the firewall security as IDS provides some extra functionality like producing alerts, logs of messages for the reference to the administrator. [2].

1.1 Background

An intrusion detection system has been identified as an effective research field of research for about three decades. Computer Security Threat Monitoring and surveillance was the first paper that was published by James Anderson working in this area in 1980. The government projects concerning the development of IDS were undertaken by Peter Neumann and Dorothy Denning during 1983 and 1986. Their research successfully developed the real time IDS that was the first model of IDS. It was popularly known as Intrusion Detection Expert System (IDES). Some experts had carried out some attacks on the internet sites, in 1980s. Some exploiting scripts and self-regulating tools were used for attacking the system. When the experts found that the number of security threats had been increasing during 2006 to 2010, the deployment of IDS for network security became a primary need for protecting the from attacks [2].

1.2 Types of IDS

1.2.1 Network-based IDS (NIDS)

Network Intrusion Detection System includes network intrusion detection capabilities. It analyses a traffic passing throughout the subnet. The traffic over the network is compared with the database of known attacks. The administrator will be sent an alert if the attack is identified. NIDS monitors the traffic going through the particular network segment or devices. NIDS collects the data as the network packets; So it is also called as packet-sniffer [2].

1.2.2 Host-based IDS (HIDS)

In Host Intrusion Detection System, the malicious activities taking place in the single host are scanned. HIDS collects logs, operations, unauthorised access, alterations and unusual changes in the configuration of the system. HIDS is deployed on the most crucial hosts containing highly essential and openly available information. Such hosts include workstations or servers [2].

1.3 IDS techniques

There are four basic techniques of IDS for two basic types of IDS (HIDS and NIDS): [3].

1.3.1 Anomaly detection

The IDS establishes a normal usage pattern and anything that widely deviates from it gets flagged is considered as possible intrusion. An anomaly is an incident that occurs on frequency less than or greater than a standard deviation from statistical point of view. Anomalies are identified by deviations from normal behaviour and any deviation from it is flagged as suspect. In this way, new types of intrusions can be identified by using new patterns in the deviation from normal usage or pattern. The drawback of using this technique is that it raises a very high false alarm and any previously unseen behaviour can also be categorised as an attack. It is designed to uncover the abnormal patterns of behaviour [3].

1.3.2 Misuse detection (Signature detection)

In this technique, there is the dataset in which each of the instances is labelled as either normal or attack and a learning algorithm is trained on the labelled data. As long as the instances are labelled appropriately; the intrusion detection model can be retrained accordingly that include new types of attack. Models of misuse

are sophisticated as they are automatically created. They can detect known attacks with great accuracy. Their disadvantage is that they cannot detect new attacks and they depend on signatures extracted by human experts. The known patterns of unauthorised behaviour are specifically used to detect and predict subsequent similar attempts. These specific patterns are called signatures. One example of signature is "three failed logins", for hostbasedintrusion detection. A specific pattern that matches a portion of network packet can be as simple as a signature for network intrusion detection. For example, an unauthorised action can be indicated by header content signatures and/or packet content signatures. Some response, alarm, or notifications should be sent to the proper authority relying on the seriousness or robustness of the signature that is triggered [3].

1.3.3 Target Monitoring

These systems look for the modification of specified files instead of actively searching for anomalies or misuse. This is designed to uncover an unauthorized action after it occurs to reverse it. The cryptographic hash is computed beforehand to check for the covert editing of files. This type of system does not require constant monitoring by the administrator, So it is the easiest to implement. It needs to compute integrity checksum hashes at whatever intervals and on either all files or just the mission/system critical files [3].

1.3.4 Stealth Probes

The attacker who chooses to carry out his/her mission for long period is detected using this technique. For example, an intruder launches an attack to know the vulnerabilities and open ports in the system for a particular period and then he waits for two months and again launches the actual attack. By collecting a variety of data throughout the system, stealth probes check for any methodical attacks over an extended period of time. In an attempt to uncover suspicious activity, this

method combines anomaly detection and misuse detection [3].

1.3.5 Hybrid based IDS

Hybrid based IDS combines all the advantages of the IDS techniques and overcomes their drawbacks. We have used this approach to improve our intrusion detection system. In our IDS, anomaly based approach is used to detect new attacks. The signature based approach is used to generate the rules for unknown attacks. The target monitoring and stealth probe are also used to identify suspicious activities.

1.4 KDDcup99 Dataset

The KDD cup99 dataset is the benchmark dataset for intrusion detection system. There are three KDD datasets, i.e., KDD corrected, nslKDD, GureKDD. There are 311029 instances in KDD corrected dataset, 125973 in nslKDD dataset and 178835 in GureKDD dataset. There are 41 features in each of the dataset, and every record is labelled as either normal or attack. The attack falls in one of the four categories: Denial of service attack (DoS), User to root attack (U2R), Remote to Local Attack (R2L), Probing Attack [4].

- Denial of service attack (DoS): In DOS attack, an attacker prevents the authorised user from consuming and accessing the services through land, back, Neptune, pod, teardrop and Smurf.

- User to root attack (U2R): The attacker attacks the victim machine with ?Buffer overflow? attack to accesses super user privileges while he has right for only local access.

- Remote to Local Attack (R2L): In this type of attack the attacker guesses or breaks the password to access the victim machine.

- Probing Attack: In this attack, an attacker attempts to know the information about the victim machine with the intent to check the vulnerability. For example, port scan.

The features are categorised into three types: Basic features, traffic features and content based features [5].

- Basic features: This category encapsulates all the attributes that can be extracted from a TCP/IP connection.
- Traffic features: The feature that are computed with respect to a window interval are included in this category. The two groups of these features are "same host" features and "same service" features.
- Content features: Some attacks like U2R and R2L are related to the data portion of the packets. Some features are needed for detecting the suspicious behaviour that is seen in these attacks. The features like the number of failed login attempts are called content features.

1.4.1 KDD corrected dataset

In KDD corrected dataset the existing irrelevant and redundant features are deleted from the dataset that results in less resource consumption, faster training and testing process, as well as maintaining high detection rates.

1.4.2 GureKDDcup Dataset

GureKDD dataset contains the information that has directly been extracted from the payload of each connection. For generating GureKDDcup dataset, the same steps are followed by the GureKDDcup capture team as that of KDD cup 99 dataset. The tcpdump files were processed with bro-ids to get each connection with its attributes. Finally, each connection of the dataset is labelled based on connections-class files provided by MIT. The size of the original dataset is 9.3 GB, and the size of 6%

dataset is 4.2 GB. Santosh et al. have tabulated the remaining information regarding attack categories, number of samples, duplicate records, their reduced rate, sample categories, a number of samples after reduction of duplicate samples, etc. in [4].

1.4.3 NSLKDD dataset

NSLKDD dataset is an attempt to solve some of the problems that are discussed in [5]. There are reasonable number of instances in the NSLKDD test dataset (22544) and train dataset (125973). So, there is no need to select randomly a small portion of the set, and the experiment can be run on the complete set. By using this dataset, consistent and comparable results can be obtained. The testing dataset includes some attacks that are absent in the training set. The more details are present in [5].

1.5 Framework of IDS using honeypot and SVM classifier

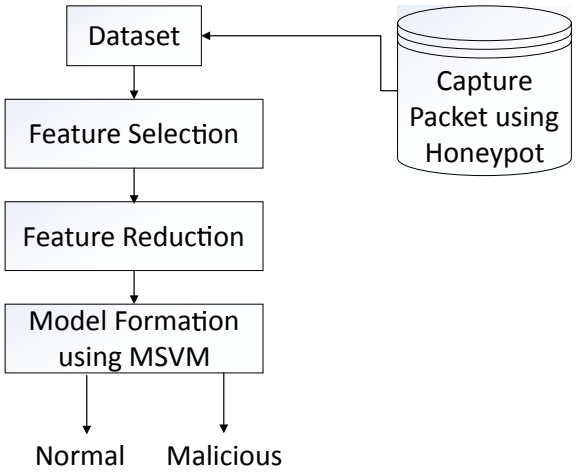

Figure 1.1: Block Diagram

1.6 Feature Selection

The dataset may contain many irrelevant and redundant attributes(features) which do not contribute to the output. But their presence affects the performance of the system. It is necessary to identify and remove such attributes. This process is called as feature selection. It not only increases the speed and accuracy of the system but makes the system work effectively. The characteristics of feature are as follows: [6]:

- Relevant: The features that are having an impact on the output are relevant attributes. They cannot be removed or replaced.
- Irrelevant: The features that do not have any impact on the output are irrelevant features.
- Redundant: whenever a feature can take the role of another, and there is no effect of removing the feature, a redundancy exists.

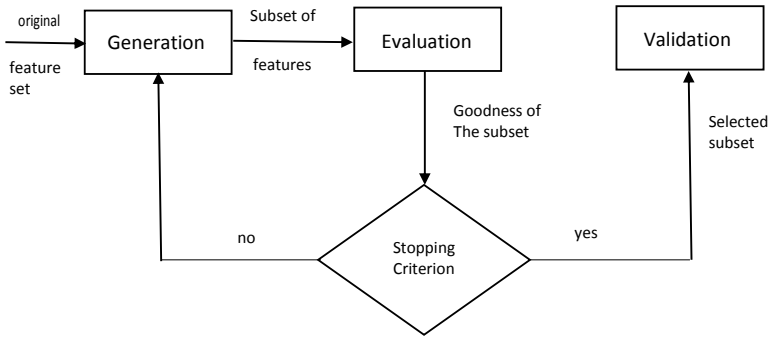

Figure 1.2: Feature subset selection

1.7 Honeypot

Honeypot is the system to deceive the attacker by providing the decoy system that seems to be highly valuable, but badly secured so that the attacker can interact with that system. The administrator can analyze the attacker's interaction with the

system and categorize that attack by which the intent of the attackers can be known as discussed in [7] [8].

If a honeypot successfully interacts with the intruder, the intruder will never know that she/he is being monitored and tricked. Most of the honeypots are installed inside firewalls through which it can be controlled in a better way, although it can also be installed outside the firewalls. A firewall restricts the traffic coming from the Internet, whereas honeypot allows the traffic on the Internet and restricts the traffic sent back from the system [7]. The parameters that are used to know the value fetched from a Honeypot are given by [9]: (i) Type of deployment of honeypot and (ii) Scenario of deployment (location of deployment i.e. behind firewall inside DMZ, in front of firewall etc).On the basis of these parameters a honeypot can act in the same way as bulgur alarm for detection of attacks, Prevention of attacks by deception and deterrence, responding to attacks by providing valuable logs regarding attack [9].

1.7.1 Areas of deployment

There are two areas of deployment of honeypot: physical honeypots and virtual honeypots. In case of physical honeypots, the original system is allowed to compromise completely by the intruder. There is a risk to the system to be damaged by the intruder. So, another approach called as a virtual honeypot that provides the attacker with a vulnerable system that is not actually the real system is used, but the attacker never knows that he is dealing with the virtual system [9].

1.7.2 Types of Honeypot

There are two types of honeypot: High Interaction Honeypot and Low Interaction Honeypot. In a high-interaction honeypot, the attacker can interact with a real system. While a low-interaction honeypots provides only some parts such as the network stack. The high interaction honeypot allows the adversary to compromise

fully the system to launch the network attack. There is a greater risk in deploying high interaction honeypot. It takes more time for analyzing the events; it may take several days to know the intent of the attacker. It needs high maintenance, so it is very hard to deploy. These are the drawbacks of a high interaction honeypot.

Due to the drawbacks and risk in the deployment of a high-interaction honeypot, we have used the low interaction honeypot. Low-interaction honeypots are used to collect the statistical data and high-level information about attack patterns. Since an attacker interacts just with a simulation, he cannot fully compromise the system. A controlled environment is constructed by Low-interaction honeypots, and thus the limited risk is involved: As the attacker cannot completely compromise the system, we do not need to worry about abuses of our low-interaction honeypots [9].

1.8 Support Vector Machine

The classification is used to achieve high accuracy for classifying maximum number of instances with a small number of training samples. Support Vector Machine is one of such classifier that can be applied to the intrusion detection dataset. The orientation of hyperplane is in such a way that the maximum distance is maintained between the sets of support vectors. This orientation determines the accuracy of classification with SVM classifier for unseen instances. For establishing the decision surface, a few training samples that are present at the edge of the class distribution of the support vectors are needed. On the other hand, if conventional maximum-likelihood classification is used, large number of training samples are required to gain accurate classification. Hence, using SVM for classification saves the training data acquisition [10].

Support vector machine classifier gives a better result for two-class classification problem [11]. It maps input vectors to a high dimensional feature space. A hyperplane separates both linear and non-linear data in two classes. The hyperplane is found with the help of support vector (training tuples) and margin (defined

by support vectors) [12]. SVMs are the successful and resilient classification algorithms [11] [13]. The SVM supports only binary classification and deals with maximizing the margin that is the minimum distance from nearest example to the separating hyperplane. The concept of SVM can be extended to multiclass classification [14] [15].

1.9 Multiclass Support Vector Machine

As SVM solves only binary class classification problem, the multiclass problem needs to be decomposed into several binary class problems. Each of the binary classifiers is applied to new data point and the frequency of number of times the point is assigned the same label is counted and the label with highest count is assigned to that point. There are several methods for the decomposition of multiclass problem [16].

1.9.1 One-verses-all

One-verses-all is also called as the winner takes all strategy. This is the simplest approach to reducing the problem of classification from k classes into k binary problems. Each problem is different from other k-1 problems. This method requires k binary classes in which we train k_th classifier with positive example belonging to class k and negative examples belonging to other k-1 classes. An unknown example is tested, and the classifier for which maximum output is produced is considered to be the winner class. That class label is assigned to that example. Although this approach is simple, its performance can be compared with more complicated approaches [17].

1.9.2 One-versus-one

For every pair of different classes, one binary classifier is constructed. In this way, the multiclass problem is broken into a series of a set of binary class problems so

that we can apply SVM model for each pair of classes. Total k(k-1)/2 classifiers are needed to classify the unknown data. The binary classifier is trained taking one class as positive and other class as negative. For a new data point x if that classifier classifies x in first class, then a vote is added to thisclass. If the classifier classifies x in the second class, the vote is added to the second class. This process is repeated for each of the k(k-1)/2 classifiers. Finally, the label of the class with maximum number of votes is assigned to the new data point x. In this way the class to which the unknown data point belongs is predicted [17] [18].

1.10 Motivation

The standard intrusion detection dataset contains 22 types of attack. The research work on machine learning and intrusion detection system have classified the attacks on two classes or five classes [2]. As the number of attacks is growing day by day, there is a need to classify the attack in exactly one of the 22 classes with maximum accuracy. Tarun et al. [1] have classified the unknown data in one of the 22 classes of attacks with an accuracy of 91.673%. The accuracy can further be improved by using different datasets and new multiclass SVM technique. The greater accuracy and speed will improve the performance of IDS. This paper provides the better solution for such type of classification.

1.11 Objective

- Removing irrelevant and redundant attributes using gini index

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{1.1}$$

  p_i is the probability that the tuple belongs to class C_i where i is the number of classes. D is the dataset.

- Logging and analysing new attacks using honeypot.

- Implementing multiclass SVM classifier for categorising attack in its specific type.

1.12 Thesis Layout

The organization of thesis is as follows —

Chapter 2:Literature Review In this chapter, the detailed study on literature on feature selection, feature reduction, honeypot and multiclass SVM.

Chapter 3:Classification of attacks using MSVM This chapter contains honeypot implementation, Methods in multiclass SVM and comparison of the whole system with existing system.

Chapter 4:Result and discussion This chapter includes the result of feature selection using Gini index, feature reduction using PCA and SVM, Detection of a particular attack using multiclass SVM.

Chapter 5:Conclusion

Chapter 2

Literature Review

In this chapter, the detailed study of literature on feature selection, feature reduction, honeypot and multiclass SVM. The dataset may contain many irrelevant and redundant attributes(features) which do not contribute to the output. But their presence affects the performance of the system. It is necessary to identify and remove such attributes. This process is called as feature selection. It not only increases the speed and accuracy of the system but makes the system work effectively. The dimensionality of the feature space is referred to as the feature reduction. Measurement of a certain aspect of an object is called as dimension. The study for methods of reducing number of dimensions describing the object is called as dimensionality reduction or feature reduction. Honeypot is the system to deceive the attacker by providing the decoy system that seems to be highly valuable, but badly secured so that the attacker can interact with that system. Multiclass Support Vector Machine is the extended form of SVM classifier that can be applied to the intrusion detection dataset to classify the attack in its particular type.

2.1 Feature Selection

Relief et al.. [19] preprocesses irrelevant features and eliminates them by using statistical methods. Based on the Euclidian distance measure, near miss and near hit

instances are found for each instance in a sample of instances that are picked randomly. An instance is chosen, and the instance that is having minimum Euclidean distance as compared to the other instances to that of the selected instance is near hit. The instance with the minimum Euclidean distance among all instances of the different class is called as the near miss. In the beginning, the weights of the features are initialized to zero, and they are updated based on the intuitive idea that a feature is more relevant if it distinguishes an instance and its near miss. The weights of all the features are compared with the threshold value. All thefeatureswith greater weights than the threshold are chosen.

Cardie et al. [20] implements an algorithm based on decision tree for feature selection as per the need. Proven [21] uses the same approach with greedy method for constructing Bayesian Network. According to Hall, the useful features are correlated with the class and not related to one another. Based on this proposition, the set of features is evaluated for their usefulness. For discovering feature subset boosted decision stumps are used [22]. According to Hall [23], the good features are correlated with the class and not related to one another. Based on this proposition, the set of features is evaluated for their usefulness. Yu and Liu [24] presented a novel approach for correlation of features and introduced a fast filter method that can remove irrelevant and redundant features.

2.2 Feature Reduction

The reduction of dimensionality of the feature space is referred to as the feature reduction. Measurement of a certain aspect of an object is called as dimension. The study for methods of reducing number of dimensions describing the object is called for dimensionality reduction. Removing irrelevant and redundant data reduces the computational cost and avoid data over-fitting [25]. It improves the quality of data for efficient data-intensive processing task such as data mining and pattern recognition. The experiments have shown that as the dimensionality increases, the

performance of the system decreases [26].

2.3 Honeypot

Honeypot is the system to deceive the attacker by providing the decoy system that seems to be highly valuable, but badly secured so that the attacker can interact with that system. The administrator can analyze the attacker's interaction with the system and categorize that attack by which the intent of the attackers can be known, as discussed in [7] [8]. The related work in honeypot is given in Table 2.1.

2.4 Multiclass SVM

Chen et al. [34] uses hierarchical SVM for clustering the classes into the binary tree. The clusters are formed by arranging the classes into the undirected graph. The weights of the edge are the Kullback-Leibler distances between every class and other class. Each node of the tree is the binary classifier SVM. Hansung Lee et.al. [35] proposes a new intrusion detection system model. This model supports advantages of both signature-based and anomaly based IDS and overcomes their disadvantages. This new IDS called multistep multiclass IDS satisfies the following necessities: 1) Speeds up detection, 2) Gives detailed information of attacks, 3) Efficient with respect to cost and learning, 4) Increases scalability of the system. It distinguishes normal and attack data and classifies it into one of the five attacks DOS, R2L, U2R and Probing attacks. Hsu et.al. [36] has proposed two methods one by considering all data at once and second is a decomposition implementation.

Srinivas Mukkamala [3] et.al. constructs SVM intrusion detection system that consists of three phases:

1. Preprocessing: an automated parser is used for processing TCP/P data which isselected randomlyand converting into machine readable format.

Table 2.1: Related Work in Honeypot

Year	Author	Type	Attack Type	Work Done
2011	Saurabh et. al [9]	l	NA	Due to the lack of capabilities of existing security devices, there is a need to study honeypot deployment and analyze tools, methods and targets of the attacker.
2006	Nguyen et al. [8]	h	NA	The purpose of this paper is to deploy a honeypot in such a way that it is well concealed from the intruder. The honeypot is deployed on Xen virtual machine with system Xebek.
2004	Kuwatly et al. [27]	h+l	NA	The dynamic honeypot approach integrates passive or active probing and virtual honeypot.
2006	Alata et al. [28]	h+l	U2R	The results based on a six months period using high interaction honeypot concludes that if the password is found to be weak it is replaced by strong one
2009	Vinu et al. [29]	NA	DOS	This paper has proposed the effective honeypot model for secured communication of authorized client and server
2009	Shujun et al. [30]	NA	Phishing	Honeypot is used to collect important information regarding attackers activity
2009	Jianwei et al. [31]	h	malware	This paper has introduced a high interaction toolkit called HoneyBow containing three tools MwFetcher, MwWatcher, MwHunter
2003	Lance et al. [32]	NA	The advance insider	This paper detects the threats done by the authorised insider
2009	Almotairi et al. [33]	l	NA	The technique for detecting new attacks using PCA with low interaction honeypot is presented in this paper

2. Training: The SVM is then trained to identify normal and attack data. 41 features are classified into two classes normal and attack. The experiment is carried out with the default regularisation parameter c=1000 and optimisation is done for 2733 iterations. The total number of misclassified data points are 6 out of 7312 training set. CPU runtime is 17.77 sec and a difference of 0.00072 is achieved. 204 support vectors with 29 at upper bound are used. The linear loss during the process was 17.78295.

3. Testing: The performance is measured for testing data. The testing set contains 6980 points with 41 features. The accuracy received is 99.50% with total runtime 1.63 sec.

According to latest research, there are a lot of attempts to improve IDS using the data mining and machine learning techniques [37]. The detection accuracy can be maximised by using machine learning algorithms. Chandrasekhar et al. proposed a four module approach for classification.

1. K-means clustering module: The training dataset is clustered into k clusters.

2. Neuro-fuzzy training module: This module trains k neural networks, Each of the data is trained with the neural network that is associated with the cluster to which that data belongs.

3. SVM training vector module: SVM classification vector is generated in this module.

4. Radial SVM classification module: Intrusion is detected by applying radial SVM, and the classification is done accordingly.

The neuro-fuzzy architecture is used to show how the classifiers are trained. It has used five classifiers for five clusters. The SVM classifier reduces the number of attributes from 34 to 6. The experimental setup uses sensitivity, specificity and

accuracy as the performance evaluation metrics.

Sensitivity is the proportion of actual positives which are correctly classified.

$$Sensitivity = \frac{TP}{TP + FN} \tag{2.1}$$

Specificity is the proportion of true negatives which are correctly classified.

$$Specificity = \frac{TN}{TN + FP} \tag{2.2}$$

The accuracy is the closeness of measurement of true values.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{2.3}$$

The accuracy is calculated for each of the four classes i.e., DOS, U2R, R2L, Probe attack. Tarun et al. [1] have focused on applying multiclass SVM classifier with the one-versus-one approach. Both misuse detection and anomaly detection systems are used to identify the attack. The KDD cup99 dataset was used as a training and testing dataset. The implementation has been done in two steps: Data preparation for training and Training and testing.

1. Data preparation for training

   (a) Tarun et al. has used LIBSVM, which accepts only alphanumeric values. So different types of alphanumeric strings need to be determined for each type of attacks in the training as well as testing dataset.

   (b) The numeric values are assigned to the strings and labels of different types of attack. In this way, KDD dataset is converted into the format accepted by LIBSVM.

   (c) Precise distribution of attack is determined.

(d) The confusion matrix is constructed and the cost per training sample is determined in the testing stage.

2. Training and testing:

(a) The Radial Basic Function (RBF) kernel option is used, and the value of C is changed and the value of gamma is set to the default of 1/41. The training is done in 23 classes including 22 attacks and one normal class.

(b) The datasets that are used for training and testing have a lot of difference in the probability distribution. So cross-validation is not used.

The classification accuracy for a multiclass having 23 classes has the accuracy of 91.6738%.

2.5 Summary

In this chapter, we have discussed how the feature selection is used in different dataset by various authors. Feature reduction is crucial for reducing the dimensionality of the dataset. Related work in honeypot and Multiclass support vector machine is mentioned in this chapter. The next chapter gives the details of the implementation of all these techniques.

Chapter 3

Classification of attacks using MSVM

As we have seen in Chapter 2, the attacks are classified into two or five categories according to the previous research. But there is a need to know the exact type of an attack. To fulfill this purpose, Multiclass SVM, which is a new technique in the field of intrusion detection, is used. As SVM solves only binary class classification problem, the multiclass problem needs to be decomposed into several binary class problems. Each of the binary classifiers is applied to new data point and the frequency of number of times the point is assigned the same label is counted and the label with highest count is assigned to that point. For data preprocessing, we are using Gini index as a feature selection technique and Principle Component Analysis as a feature reduction technique. Honeypot system is also included for collecting more information regarding new attacks. This system will improve the intrusion detection system.

3.1 Feature selection using Gini index

Anomaly detection refers to finding out the abnormal pattern of traffic or abnormal behavior from network or system. The dataset used for intrusion detection needs to

be preprocessed for better results. For feature ranking, Gini index is used to find the ranking of each of the 41 features of KDDcup99 dataset for getting most relevant features [12].

$$Gini(D) = 1 - \sum_{i=1}^{m} p_i^2 \tag{3.1}$$

p_i is the probability that the tuple belongs to partition i where i is the number of partitions. D is the dataset.

The binary partitioning is used for partitioning the dataset. The dataset is divided into 2 parts D_1 and D_2. The gini index for attribute A in the dataset is calculated by using following equation:

$$Gini_A(D) = \frac{|D_1|}{|D|} Gini(D_1) + \frac{|D_2|}{|D|} Gini(D_2) \tag{3.2}$$

The reduction in impurity (or the loss of information)after partition on attribute A is given by:

$$\Delta Gini(A) = Gini(D) - Gini_A(D) \tag{3.3}$$

The attribute which gives maximum reduction in the impurity is selected as the most relevent attribute.

3.2 Feature Reduction

The principal component analysis is used for reducing the dimensionality of the dataset. The basic steps are as follows:

- First the input data is normalised to make each attribute fall in the given range. The normalised data controls the dominance of large domain attributes over the small domain attribute.
- The k orthonormal vectors are computed to provide the basis of normalised input data. Each vector points in a direction perpendicular to the other. They are called as principal components.

- The principal components are sorted in order of decreasing significance or strength. They serve as a new set of axes that provide essential information about axes in a way such that first axes gives maximum variance, the second gives the next highest variance and so on. In this way the first few principal components that contribute more variance are considered and the remaining are discarded.

3.3 Honeypot Configuration

The honeypot is configured on the virtual system like Vmware. In low interaction honeypot, there are individual fingerprint files that contain the information about how the particular operating system will respond. For example, if we want to show the attacker that we are running Windows XP operating system, it will react with certain characteristics, which will be used by the honeypot to respond to the attacker. The attacker will think that he is working with the Windows XP operating system, but he will never know that he is dealing with the virtual operating system. The few of the essential features of honeyd are creation, setting, binding and adding. In the configuration process, we are going to create a template with some name or default:

create$< template - name >$

create default

dynamic$< template - name >$

Then we set the personality of the honeypot, i.e, the operating system and mention certain protocol or action such as reset, block or open.

$set < templatename > personality < personality - name >$

$set < templatename > default < proto > action < action >$

We are adding the particular template along with protocol name, port number and action.

$add < template - name >< proto > port < port - number >< action >$

Fig.3.1 shows the working model of IDS and honeypot together. The intrusion detection System redirects the attacker to the honeypot when the malicious activity is detected. The intruder interacts with the honeypot and tries to know its vulnerabilities and open ports. The honeypot allows to gain access to the limited resources of the system so that it should not make any harm to the necessary files and resources. The honeypot logs the attack activities of the particular intruder. This log file is then used to create new rules that are further added to the list of already generated rules. Once this is done, when the same type of behavior occurs next time, this is directly considered as attack and there is no need to redirect that intruder to the honeypot. In this way, the novel attacks can be detected by the intrusion detection system.

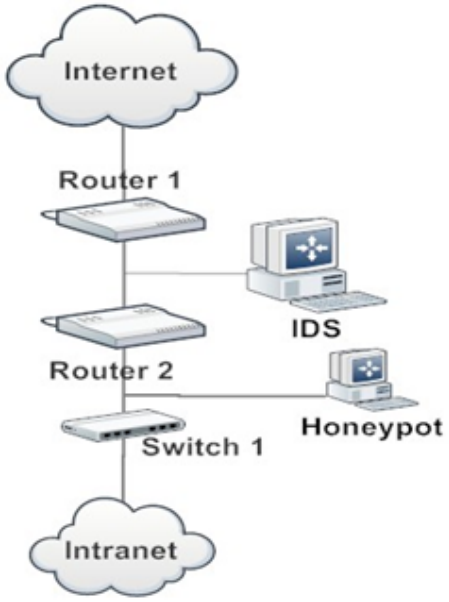

Figure 3.1: The working model of IDS and honeypot

3.4 Classification using multiclass SVM

We are using one against one method of multiclass support vector machine. The cross validation method is also used. We are using three intrusion detection datasets; KDD corrected dataset, NSLKDD dataset and Gure KDD dataset for training and testing purpose. The details are as mentioned in the table 3.1 below:

Table 3.1: The details of datasets

Dataset	Number of instances for training	Number of instances for testing	Number of instances correctly classified	Accuracy
KDD corrected	77291	311029	284421	91.445%
NSL KDD	47736	125973	118447	94.025%
Gure KDD	160904	178810	177283	99.146%

The confusion matrices for KDD corrected dataset, Gure KDD dataset and NSL KDD dataset are given in Table 5.1, Table 5.3 and Table 5.2 respectively. We have removed some rows and columns from the confusion matrix because they were having all zero values.

3.5 Comparison

It is necessary to detect the attack by its particular type with greater accuracy. The data mining techniques are very useful for such classification. We have used multiclass support vector machine approach for the classification of various unknown attacks. The comparison of existing classification and new classification is tabulated as follows:

Table 3.2: The comparison of multiclass classification

	New system	Existing System
Datasets	KDD corrected, Gure KDD, NSL KDD	KDD cup99
configuration	Intel i7, 3.4 GHz,8 GB RAM, Windows 8.1, 64-bit OS	Pentium III - 933 MHz processor with 256 MB of RAM
Accuracy of detection	91.445% using KDD corrected dataset, 94.025% using NSL KDD dataset, 99.146% using Gure KDD dataset	91.673% using KDD cup99 dataset

The existing research [1] failed to do that, and the accuracy of detection of attack was 91.673%. The old dataset KDD cup99 was used for evaluation. We have used three datasets KDD corrected dataset, NSLKDD dataset and Gure KDD dataset for training and testing. By using Gure KDD dataset, we have got the maximum accuracy, i.e., 99.146%. By using NSL KDD dataset, we got the accuracy of 94.025% and the accuracy of 91.445% using KDD corrected dataset.In earlier work, the machine configuration was as follows: Pentium III - 933 MHz processor with 256 MB of RAM. We have also calculated the confusion matrices for all the three datasets. Our system configuration is Intel i7, 3.4 GHz,8 GB RAM, Windows 8.1, 64-bit OS, and we have used Matlab 2015a. We have also calculated the confusion matrices for all the three datasets. After calculating confusion matrices, we have found that some of the rows and columns contain all zero values. So we have removed such rows and columns.

3.6 Summary

In this chapter, we have described our proposed work on feature selection to select most relevant features by using Gini index. Feature reduction using PCA with SVM classifier is very useful for dimensionality reduction. Honeypot captures the new attacks for analysis. The multiclass SVM classifier classifies the attacks in one of the particular types of attacks. In the next chapter, the results of all the proposed work are presented.

Chapter 4

Result and discussion

4.1 Feature selection using Gini Index

The Gini index method is used for finding ranks of each feature in the KDDcorrected dataset, nslKDD dataset and Gure dataset. Rank 1 indicates least relevant feature.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	**Ranks**	1	2	3	4	5	6	7	8	9	10	11	12	13
2	kdd corrected	5	4	6	28	29	33	12	32	22	34	3	39	40
3	nslkdd	5	6	4	27	28	32	31	12	33	37	23	24	36
4	GureKdd	1	35	3	37	23	38	22	6	36	30	10	8	25

Figure 4.1: Gini Result-1

O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB
14	15	16	17	18	19	20	21	22	23	24	25	26	27
26	27	31	36	35	37	24	38	25	30	23	1	2	10
21	40	3	30	35	34	29	39	25	38	26	2	22	1
31	32	27	24	5	26	17	21	18	4	13	11	28	19

Figure 4.2: Gini Result-2

AC	AD	AE	AF	AG	AH	AI	AJ	AK	AL	AM	AN	AO
28	29	30	31	32	33	34	35	36	37	38	39	40
11	21	13	19	18	16	9	7	20	17	14	8	15
8	16	19	10	13	20	17	15	14	18	9	11	7
34	33	20	40	12	29	7	16	39	2	9	14	15

Figure 4.3: Gini Result-3

4.2 Feature Reduction

The feature reduction using PCA along with SVM classifier gives the following result:

File Edit Text Go Tools Debug Desktop Window Help

Stack: Base

	Dimension	TP	TN	FP	FN	Precision	Specificity	Sensitivity	Accuracy
2	40	11731	12	113	13336	0.99	0.99	1.00	99.50
3	39	11731	12	113	13336	0.99	0.99	1.00	99.50
4	38	11731	12	113	13336	0.99	0.99	1.00	99.50
5	37	11731	12	113	13336	0.99	0.99	1.00	99.50
6	36	11731	12	113	13336	0.99	0.99	1.00	99.50
7	35	11732	11	113	13336	0.99	0.99	1.00	99.51
8	34	11732	11	113	13336	0.99	0.99	1.00	99.51
9	33	11732	11	113	13336	0.99	0.99	1.00	99.51
10	32	11732	11	113	13336	0.99	0.99	1.00	99.51
11	31	11732	11	113	13336	0.99	0.99	1.00	99.51
12	30	11731	12	113	13336	0.99	0.99	1.00	99.50
13	29	11731	12	113	13336	0.99	0.99	1.00	99.50
14	28	11732	11	113	13336	0.99	0.99	1.00	99.51
15	27	11732	11	113	13336	0.99	0.99	1.00	99.51
16	26	11731	12	113	13336	0.99	0.99	1.00	99.50
17	25	11730	13	117	13332	0.99	0.99	1.00	99.48
18	24	11729	14	116	13333	0.99	0.99	1.00	99.48
19	23	11728	15	115	13334	0.99	0.99	1.00	99.48
20	22	11729	14	115	13334	0.99	0.99	1.00	99.49
21	21	11729	14	123	13326	0.99	0.99	1.00	99.46
22	20	11727	16	122	13327	0.99	0.99	1.00	99.45
23	19	11727	16	122	13327	0.99	0.99	1.00	99.45
24	18	11727	16	124	13325	0.99	0.99	1.00	99.44
25	17	11724	19	131	13318	0.99	0.99	1.00	99.40
26	16	11715	28	146	13303	0.99	0.99	1.00	99.31
27	15	11707	36	152	13297	0.99	0.99	1.00	99.25
28	14	11684	59	169	13280	0.99	0.99	0.99	99.09
29	13	11669	74	179	13270	0.98	0.99	0.99	99.00

Figure 4.4: Feature reduction

4.3 Logs in Honeypot

For analysing new attacks, it is necessary to get more information about the attack. For this purpose, we have used the honeypot system to communicate with the attacker. The honeypot is configured in such a way that it logs the activities of attack that can be used later. The logs generated by honeypot are as shown below:

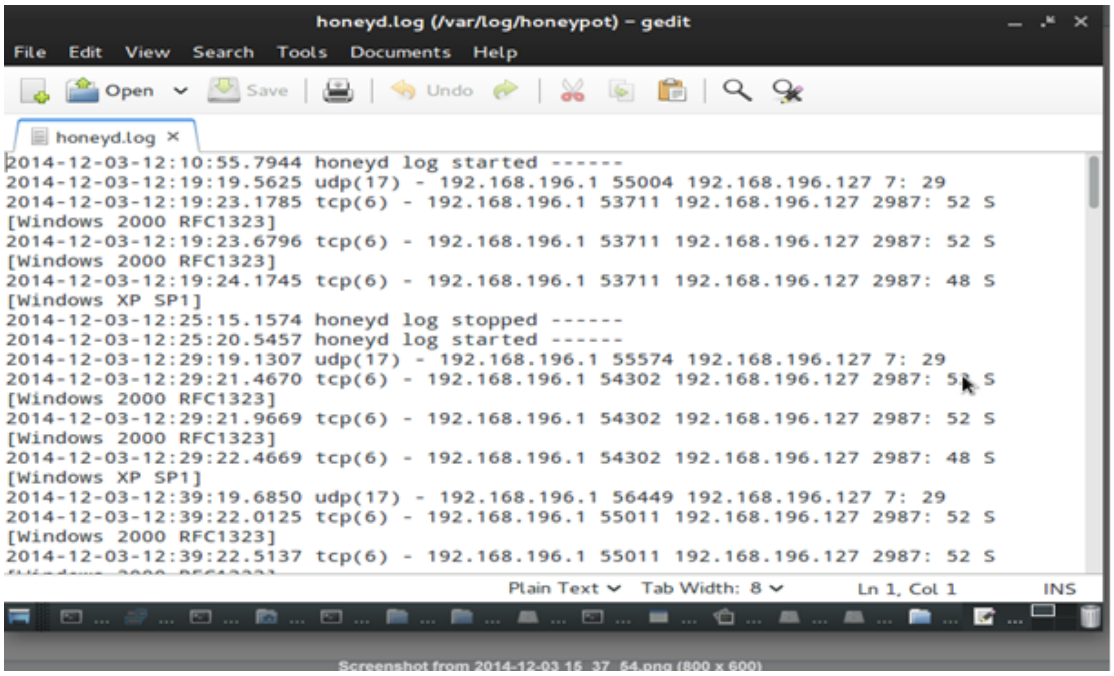

Figure 4.5: Honeypot configuration

4.4 Result of implementing Multiclass SVM

The three datasets are used for training and testing of multiclass support vector machine classifier: KDD corrected dataset, NSL KDD dataset and Gure KDD dataset. The results are mensioned in the subsequent sections.

4.4.1 Attackwise Accuracy of Datasets

The multiclass SVM classifier classifies the attack with its particular type. There are total 23 attacks in NSLKDD dataset, 38 attacks in KDD corrected dataset and 28 attacks in Gure KDD dataset. The detection accuracy of the classifier according to each attack in the respective datasets is tabulated as follows:
Table 4.1 presents attackwise accuracy of NSL KDD dataset.
Table 4.2 presents attackwise accuracy of KDD corrected dataset.
Table 4.3 presents attackwise accuracy of Gure KDD dataset.

4.4.2 Confusion matrices

The confusion matrices are calculated for three datasets KDD corrected dataset, Gure KDD dataset, NSL KDD dataset are as shown below:
Table 5.1 presents confusion matrix of KDD corrected dataset.
Table 5.3 presents confusion matrix of Gure KDD dataset.
Table 5.2 presents confusion matrix of NSL KDD dataset.

4.4.3 ROC curve for multiclass classification

A graphical approach for displaying balance between TPR(true positive rate) and FPR(false positive rate) of a classifier is called as a reciever operating characteristic curve. The critical points along ROC curve are interpreted as follows:
(TPR=0,FPR=0): Model predicts every instance to be negative class
(TPR=1,FPR=1): Model predicts every instance to be positive class
(TPR=0,FPR=0): Model predicts correctly. So it is the ideal model
The model is good if it is located close to the upper left corner, because the diagonal represents random guesses as it connects points (TPR=0,FPR=0) and (TPR=1,FPR=1). The ROC curve for datasets KDD corrected, NSL KDD and

Gure KDD are as follows:

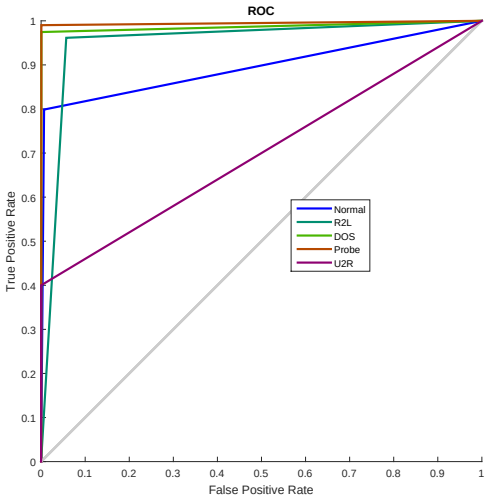

Figure 4.6: ROC curve for KDD corrected dataset

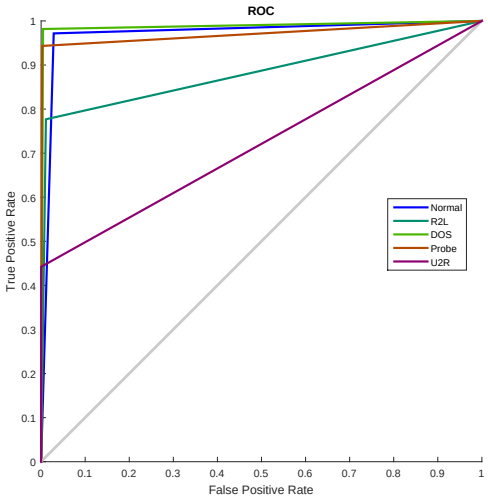

Figure 4.7: ROC curve for NSL KDD dataset

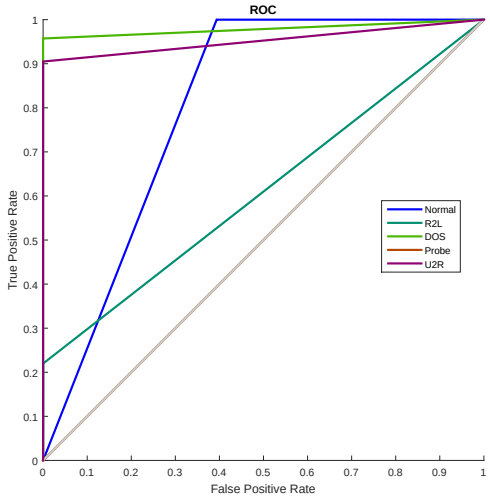

Figure 4.8: ROC curve for Gure KDD dataset

4.5 Summary

This chapter gives the detailed information of all the results regarding Feature selection using Gini index, how the features dimensionality is reduced using PCA with SVM. The attack information is logged using honeypot. The attacks can be classified into its particular type using multiclass SVM classifier. The results are presented in the tables of attack wise accuracy of datasets and confusion matrices.

Table 4.1: NSL KDD dataset

Sr no.	Attack Name	Number of instances	Accuracy
1	Normal	67343	97.68%
2	Neptune	41214	99.97%
3	Warezclient	890	55.28%
4	Ipsweep	3599	95.83%
5	Portsweep	2931	97.30%
6	Teardrop	892	99.10%
7	Nmap	1493	80.98%
8	Back	956	20.92%
9	Smurf	2646	97.51%
10	Satan	3633	91.52%
11	Warezmaster	20	95.00%
12	Buffer_overflow	30	0.00%
13	Ftp_write	8	0.00%
14	Guess_passwd	53	96.23%
15	Imap	11	0.00%
16	Land	18	94.44%
17	Loadmodule	9	0.00%
18	Multihop	7	0.00%
19	Phf	4	0.00%
20	Pod	201	51.74%
21	Rootkit	10	0.00%
22	Spy	2	0.00%
28	Perl	3	0.00%

Table 4.2: KDD corrected dataset

Sr no.	Attack Name	Number of instances	Accuracy
1	Normal	9711	95.55%
2	Neptune	4657	99.83%
4	Ipsweep	141	97.16%
5	Portsweep	157	96.82%
6	Teardrop	12	66.67%
7	Nmap	73	100.00%
8	Back	359	18.11%
9	Smurf	665	100.00%
10	Satan	735	98.91%
11	Warezmaster	944	90.47%
12	Buffer_overflow	20	35.00%
13	Ftp_write	3	0.00%
14	Guess_passwd	1231	79.29%
15	Imap	1	0.00%
16	Land	7	100.00%
17	Loadmodule	2	0.00%
18	Multihop	18	0.00%
19	Phf	2	0.00%
20	Pod	41	80.49%
21	Rootkit	13	0.00%
23	Apache2	737	99.46%
24	Httptunnel	133	93.98%
25	Mailbomb	293	75.43%
26	Mscan	996	92.67%
27	Named	17	0.00%
28	Perl	2	0.00%
29	Processtable	685	99.71%
30	Ps	15	0.00%
31	Saint	319	0.00%
32	Sendmail	14	0.00%
33	Snmapgetattack	178	0.00%
34	Snmapguess	331	96.68%
35	Sqlattack	2	0.00%
36	Udpstorm	2	0.00%
37	Worm	2	0.00%
38	Xlock	9	0.00%
39	Xsnoop	4	0.00%
40	Xterm	13	0.00%

Table 4.3: Gure KDD dataset

Sr no.	Attack Name	Number of instances	Accuracy
1	Anomaly	9	0.00%
2	Dict	879	98.29%
3	Dict_simple	1	0.00%
4	Eject	11	0.00%
5	Eject_fail	1	0.00%
6	Ffb	10	0.00%
7	Ffb_clear	1	0.00%
8	Format	6	0.00%
9	Format_clear	1	0.00%
10	Format_fail	10	0.00%
11	Ftp_write	8	0.00%
12	Guest	50	98.00%
13	Imap	7	0.00%
14	Land	35	100.00%
15	Load_clear	1	0.00%
16	Multihop	9	0.00%
17	Normal	174873	99.99%
18	Perl_clear	1	0.00%
19	Perlmagic	4	0.00%
20	Phf	5	0.00%
21	Rootkit	29	0.00%
22	Spy	2	0.00%
23	Syslog	4	0.00%
24	Teardrop	1085	99.17%
25	Warez	1	0.00%
26	Warezclient	1749	21.73%
27	Warezmaster	19	78.95%
28	Loadmodule	8	0.00%

Chapter 5

Conclusion

As the attacks and information threats are increasing rapidly there is a need for an improved intrusion detection system that can cope with the situation. We have designed an intrusion detection system working along with honeypots and multiclass support vector machine classifier. The primary objective of the honeypot is to collect intense attack patterns and decode it into human understandable format. We have implemented a virtual honeypot using honeyd which is installed on Ubuntu 14 machine and the attack patterns are captured whenever recommended by the IDS. The well-known probe attacking tools are used for attacking the system by us. The packets captured by the honeypot is decoded and converted into csv format for subsequent analysis. A multiclass support vector machine classifier is used for classification of attacks on intrusion datasets. Three benchmark datasets namely KDD corrected, NSL KDD and GureKDD are used for training and testing the model. The MSVM model is implemented using LIBSVM class under Matlab 2015a. Cross validation method is applied to the datasets to select proper subset of training and testing instances. The model can determine a particular known type of attack when the unknown instances need to be classified. This method provides better detection accuracy and reduces the complexity of the model.

Scope for Further Research

The model can be improved training and testing with the new upcoming intrusion detection datasets with more number of attacks and it can also be optimized by using good optimization algorithms.

Bibliography

[1] Tarun Ambwani. Multi class support vector machine implementation to intrusion detection. In *Neural Networks, 2003. Proceedings of the International Joint Conference on*, volume 3, pages 2300–2305. IEEE, 2003.

[2] Hussain Ahmad Madni Uppal, Memoona Javed, and MJ Arshad. An overview of intrusion detection system (ids) along with its commonly used techniques and classifications. *database*, 19:20.

[3] Vera Marinova-Boncheva. A short survey of intrusion detection systems. *Problems of Engineering Cybernetics and Robotics*, 58:23–30, 2007.

[4] Santosh Kumar Sahu, Sauravranjan Sarangi, and Sanjaya Kumar Jena. A detail analysis on intrusion detection datasets. In *Advance Computing Conference (IACC), 2014 IEEE International*, pages 1348–1353. IEEE, 2014.

[5] Mahbod Tavallaee, Ebrahim Bagheri, Wei Lu, and Ali-A Ghorbani. A detailed analysis of the kdd cup 99 data set. In *Proceedings of the Second IEEE Symposium on Computational Intelligence for Security and Defence Applications 2009*, 2009.

[6] Hai Thanh Nguyen, Katrin Franke, and Slobodan Petrovic. Feature extraction methods for intrusion detection systems. *Threats, Countermeasures, and Advances in Applied Information Security*, page 23, 2012.

[7] Niels Provos and Thorsten Holz. *Virtual honeypots: from botnet tracking to intrusion detection*. Pearson Education, 2007.

[8] Nguyen Anh Quynh and Yoshiyasu Takefuji. Towards an invisible honeypot monitoring system. In *Information Security and Privacy*, pages 111–122. Springer, 2006.

[9] Saurabh Chamotra, JS Bhatia, Raj Kamal, and AK Ramani. Deployment of a low interaction honeypot in an organizational private network. In *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, pages 130–135. IEEE, 2011.

[10] A Mathur and GM Foody. Multiclass and binary svm classification: Implications for training and classification users. *Geoscience and Remote Sensing Letters, IEEE*, 5(2):241–245, 2008.

[11] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[12] Jiawei Han, Micheline Kamber, and Jian Pei. *Data mining, southeast asia edition: Concepts and techniques*. Morgan kaufmann, 2006.

[13] Christopher JC Burges. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, 2(2):121–167, 1998.

[14] Yoonkyung Lee, Yi Lin, and Grace Wahba. Multicategory support vector machines: Theory and application to the classification of microarray data and satellite radiance data. *Journal of the American Statistical Association*, 99(465):67–81, 2004.

[15] Erin J Bredensteiner and Kristin P Bennett. Multicategory classification by support vector machines. In *Computational Optimization*, pages 53–79. Springer, 1999.

[16] Erin L Allwein, Robert E Schapire, and Yoram Singer. Reducing multiclass to binary: A unifying approach for margin classifiers. *The Journal of Machine Learning Research*, 1:113–141, 2001.

[17] Mohamed Aly. Survey on multiclass classification methods. *Neural Netw*, pages 1–9, 2005.

[18] Kai-Bo Duan and S Sathiya Keerthi. Which is the best multiclass svm method? an empirical study. In *Multiple Classifier Systems*, pages 278–285. Springer, 2005.

[19] Igor Kononenko, Edvard Šimec, and Marko Robnik-Šikonja. Overcoming the myopia of inductive learning algorithms with relieff. *Applied Intelligence*, 7(1):39–55, 1997.

[20] Claire Cardie. Using decision trees to improve case-based learning. In *Proceedings of the tenth international conference on machine learning*, pages 25–32, 1993.

[21] Moninder Singh and Gregory M Provan. Efficient learning of selective bayesian network classifiers. 1995.

[22] Sanmay Das. Filters, wrappers and a boosting-based hybrid for feature selection. In *ICML*, volume 1, pages 74–81. Citeseer, 2001.

[23] Mark A Hall. *Correlation-based feature selection for machine learning*. PhD thesis, The University of Waikato, 1999.

[24] Lei Yu and Huan Liu. Feature selection for high-dimensional data: A fast correlation-based filter solution. In *ICML*, volume 3, pages 856–863, 2003.

[25] Andrew Y Ng. Preventing" overfitting" of cross-validation data. In *ICML*, volume 97, pages 245–253, 1997.

[26] Richard Bellman, Richard Ernest Bellman, Richard Ernest Bellman, and Richard Ernest Bellman. *Adaptive control processes: a guided tour*, volume 4. Princeton university press Princeton, 1961.

[27] Iyad Kuwatly, Malek Sraj, Zaid Al Masri, and Hassan Artail. A dynamic honeypot design for intrusion detection. In *Pervasive Services, 2004. ICPS 2004. IEEE/ACS International Conference on*, pages 95–104. IEEE, 2004.

[28] Eric Alata, Vincent Nicomette, Marc Dacier, Matthieu Herrb, et al. Lessons learned from the deployment of a high-interaction honeypot. *arXiv preprint arXiv:0704.0858*, 2007.

[29] Vinu V Das. Honeypot scheme for distributed denial-of-service. In *Advanced Computer Control, 2009. ICACC'09. International Conference on*, pages 497–501. IEEE, 2009.

[30] Shujun Li and Roland Schmitz. *A novel anti-phishing framework based on honeypots*. IEEE, 2009.

[31] Jianwei Zhuge, Thorsten Holz, Xinhui Han, Chengyu Song, and Wei Zou. Collecting autonomous spreading malware using high-interaction honeypots. In *Information and Communications Security*, pages 438–451. Springer, 2007.

[32] Lance Spitzner. Honeypots: Catching the insider threat. In *Computer Security Applications Conference, 2003. Proceedings. 19th Annual*, pages 170–179. IEEE, 2003.

[33] Saleh Almotairi, Andrew Clark, George Mohay, and Jacob Zimmermann. A technique for detecting new attacks in low-interaction honeypot traffic. In *Internet Monitoring and Protection, 2009. ICIMP'09. Fourth International Conference on*, pages 7–13. IEEE, 2009.

[34] Yangchi Chen, Melba M Crawford, and Joydeep Ghosh. Integrating support vector machines in a hierarchical output space decomposition framework. In *Geoscience and Remote Sensing Symposium, 2004. IGARSS'04. Proceedings. 2004 IEEE International*, volume 2, pages 949–952. IEEE, 2004.

[35] Hansung Lee, Jiyoung Song, and Daihee Park. Intrusion detection system based on multiclass svm. In *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, pages 511–519. Springer, 2005.

[36] Chih-Wei Hsu and Chih-Jen Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415–425, 2002.

[37] AM Chandrasekhar and K Raghuveer. Intrusion detection technique by using k-means, fuzzy neural network and svm classifiers. In *Computer Communication and Informatics (ICCCI), 2013 International Conference on*, pages 1–7. IEEE, 2013.

Dissemination

Conference

1. Kanchan Shendre, Santosh Kumar Sahu, Ratnakar Dash, Sanjay Kumar Jena, Learning probe attack patterns with Honeypots, *The Third International Conference on Advanced Computing, Networking, and Informatics*, June 2015 (Accepted)

2. Kanchan Shendre, Santosh Kumar Sahu, Ratnakar Dash, Sanjay Kumar Jena, Multiclass SVM classification for intrusion detection, *The Fourth IEEE International Conference on Advances in Computing, Communications and Informatics* , August 2015 (Communicated)

Appendix

Table 5.1: KDD corrected confusion matrix

0	1	2	3	5	6	8	9	11	12	15	16	17	20	21	22	25	26	28	30	34	38
1	790	2	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
2	16	98	0	0	0	0	0	0	0	0	0	984	0	0	0	0	0	0	0	0	0
3	0	0	12	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	4	0
4	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	1	0
5	2	0	0	3959	1	0	0	1	0	0	0	401	0	0	3	0	0	0	0	0	0
6	0	0	1	0	150	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0
7	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	0	0	0	0	0	296	0	0	0	0	0	5	0	0	0	2	0	3	0	0	0
9	0	0	0	0	0	0	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0
10	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
11	0	0	0	4421	0	0	0	579	0	0	0	0	0	0	0	0	0	0	0	0	0
12	0	0	0	1	6	0	0	0	1039	0	0	4	0	0	3	0	0	0	0	0	0
13	0	0	0	1	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	7	57989	0	5	0	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0	0	0	84	0	0	0	0	0	0	0	0	0	0
17	1	9	1	46	2	1	0	15	4	5	0	48539	9	65	2	0	82	2	11707	103	0

Table 5.1: KDD corrected confusion matrix continued.....

	1	2	3	5	6	8	9	11	12	15	16	17	20	21	22	25	26	28	30	34	38
18	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
20	0	0	0	0	0	0	0	0	0	0	0	13	73	0	0	0	0	1	0	0	0
21	0	0	0	0	23	0	0	0	0	0	0	1	0	330	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0	3	0	0	0	0	0	756	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	0	14	0	0	0	0	0	0	0	2	0
24	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	1
25	0	0	0	0	0	0	0	0	1	0	0	17	0	0	0	104	614	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	9	0	0	0	0	1624	0	0	0	0
27	0	0	2	2	0	0	0	0	0	0	0	9	0	0	0	0	0	0	0	4	0
28	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	164090	0	0	0
29	0	0	0	0	0	0	0	0	0	0	0	106	0	0	0	0	0	0	7635	0	0
30	0	0	0	0	0	0	0	0	0	0	0	10	0	0	0	1	0	0	2395	0	0
31	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0	0	0	0	12	0	0	0	0	0	0	0	0	0
33	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
34	0	0	0	0	0	0	0	0	0	0	0	97	0	0	0	0	0	0	0	1505	0
35	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
36	0	0	0	1	0	0	0	0	0	0	0	5	0	0	3	0	0	0	0	0	0
37	0	0	2	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0	0	0	0	11	0	0	0	0	0	0	0	2	0

Table 5.2: NSLKDD confusion matrix

0	1	2	3	4	5	6	7	8	9	10	11	12	14	16	20	23	24	25	26	28	29
1	65781	4	116	80	2	0	0	30	151	52	423	1	345	7	0	21	37	63	16	11	203
2	4	41200	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	9	0	0
3	380	0	492	0	0	0	0	0	0	0	18	0	0	0	0	0	0	0	0	0	0
4	111	0	0	3449	0	0	39	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	41	5	0	0	2852	0	0	0	0	0	0	0	0	0	0	0	27	2	4	0	0
6	7	0	0	0	0	884	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
7	271	0	0	13	0	0	1209	0	0	0	0	0	0	0	0	0	0	0	0	0	0
8	287	0	0	0	2	0	0	0	0	3325	0	0	1	0	0	0	13	0	5	0	0
9	66	0	0	0	0	0	0	0	2580	0	0	0	0	0	0	0	0	0	0	0	0
10	97	0	0	0	0	0	0	0	0	0	0	0	0	0	104	0	0	0	0	0	0
11	749	0	2	0	0	0	0	200	0	0	0	0	0	0	0	5	0	0	0	0	0
12	2	0	0	0	0	0	0	0	0	0	0	0	51	0	0	0	0	0	0	0	0
13	4	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0
14	2	0	0	0	0	0	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0
15	5	0	1	0	0	0	0	0	0	0	0	2	1	0	0	0	0	0	0	1	0
16	26	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
17	10	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	0	0	0	0	0	17	0	0	0	0	0	0	0
19	8	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	1	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0
21	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
22	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
27	1	0	0	0	0	0	0	0	0	0	19	0	0	0	0	0	0	0	0	0	0

Table 5.3: Gure KDD confusion matrix

0	2	12	14	17	24	26	27
1	0	0	0	9	0	0	0
2	864	0	0	15	0	0	0
3	1	0	0	0	0	0	0
4	0	1	0	10	0	0	0
5	0	0	0	1	0	0	0
6	0	0	0	10	0	0	0
7	0	0	0	1	0	0	0
8	0	0	0	6	0	0	0
9	0	0	0	1	0	0	0
10	0	0	0	1	0	0	0
11	0	0	0	8	0	0	0
12	0	49	0	1	0	0	0
13	0	0	0	7	0	0	0
14	0	0	35	0	0	0	0
15	0	0	0	1	0	0	0
16	0	0	0	9	0	0	0
17	3	0	2	174864	0	3	1
18	0	0	0	1	0	0	0
19	0	0	0	4	0	0	0
20	0	0	0	5	0	0	0
21	1	0	0	28	0	0	0
22	0	0	0	2	0	0	0
23	0	0	0	4	0	0	0
24	0	0	0	9	1076	0	0
25	0	0	0	1	0	0	0
26	0	0	0	1369	0	380	0
27	0	0	0	4	0	0	15
28	0	0	0	8	0	0	0